In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

OBJECTFOLDER 2.0: A Multisensory Object Dataset for Sim2Real Transfer

Ruohan Gao[1*] Zilin Si[2*] Yen-Yu Chang[1*] Samuel Clarke[1]
Jeannette Bohg[1] Li Fei-Fei[1] Wenzhen Yuan[2] Jiajun Wu[1]
[1]Stanford Univeristy [2]Carnegie Mellon University

Abstract

Objects play a crucial role in our everyday activities. Though multisensory object-centric learning has shown great potential lately, the modeling of objects in prior work is rather unrealistic. OBJECTFOLDER *1.0 is a recent dataset that introduces 100 virtualized objects with visual, acoustic, and tactile sensory data. However, the dataset is small in scale and the multisensory data is of limited quality, hampering generalization to real-world scenarios. We present* OBJECTFOLDER *2.0, a large-scale, multisensory dataset of common household objects in the form of implicit neural representations that significantly enhances* OBJECTFOLDER *1.0 in three aspects. First, our dataset is 10 times larger in the amount of objects and orders of magnitude faster in rendering time. Second, we significantly improve the multisensory rendering quality for all three modalities. Third, we show that models learned from virtual objects in our dataset successfully transfer to their real-world counterparts in three challenging tasks: object scale estimation, contact localization, and shape reconstruction.* OBJECTFOLDER *2.0 offers a new path and testbed for multisensory learning in computer vision and robotics. The dataset is available at* https://github.com/rhgao/ObjectFolder.

1. Introduction

Our everyday activities involve perception and manipulation of a wide variety of *objects*. For example, we begin the morning by first turning off the alarm clock on the nightstand, slowly waking up. Then we may put some bread on a plate and enjoy our breakfast with a fork and knife to kick off the day. Each of these objects has very different physical properties—3D shapes, appearance, and material types, leading to their distinctive sensory modes: the alarm clock looks round and glossy, the plate clinks when struck with the fork, the knife feels sharp when touched on the blade.

However, prior work on modeling real-world objects is rather limited and unrealistic. In computer vision, objects are often modeled in 2D with the focus of identifying and locating them in static images [15, 24, 39]. Prior works on shape modeling build 3D CAD models of objects [11, 72], but they tend to focus purely on geometry, and the visual textures of the objects are of low-quality. Moreover, most works lack the full spectrum of physical object properties and focus on a single modality, mostly vision.

Our goal is to build a large dataset of realistic and multisensory 3D object models such that learning with these virtualized objects can generalize to their real-world counterparts. As shown in Fig. 1, we leverage existing high-quality scans of real-world objects and extract their physical properties including visual textures, material types, and 3D shapes. Then we simulate the visual, acoustic, and tactile data for each object based on their object intrinsics, and use an implicit neural representation network—*Object File*—to

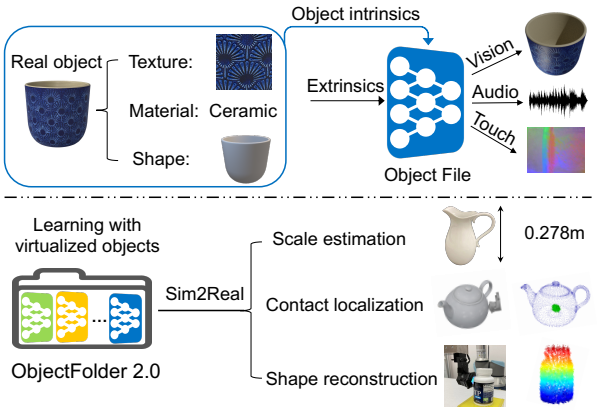


Figure 1. OBJECTFOLDER 2.0 contains 1,000 implicitly represented objects each containing the complete multisensory profile of a real object. We virtualize each object by encoding its intrinsics (texture, material type, and 3D shape) with an *Object File* implicit neural representation. Then we can render its visual appearance, impact sound, and tactile readings based on any extrinsic parameters. We successfully transfer the models learned from our virtualized objects to three challenging tasks on their real-world counterparts. This opens a new path for multisensory learning in computer vision and robotics, where OBJECTFOLDER 2.0 serves as a rich and realistic object repository for training real-world models.

*indicates equal contribution.

encode the simulated multisensory data. If the sensory data is realistic enough, models learned with these virtualized objects can then be transferred to real-world tasks involving these objects.

To this end, we introduce OBJECTFOLDER 2.0, a large dataset of implicitly represented multisensory replicas of real-world objects. It contains 1,000 high-quality 3D objects collected from online repositories [1, 2, 10, 14]. Compared with OBJECTFOLDER 1.0[1] that is slow in rendering and of limited quality in multisensory simulation, we improve the acoustic and tactile simulation pipelines to render more realistic multisensory data. Furthermore, we propose a new implicit neural representation network that renders visual, acoustic, and tactile sensory data all in real-time with state-of-the-art rendering quality. We successfully transfer models learned on our virtualized objects to three challenging real-world tasks, including object scale estimation, contact localization, and shape reconstruction.

OBJECTFOLDER 2.0 enables many applications, including 1) multisensory learning with vision, audio, and touch; 2) robot grasping of diverse real objects on various robotic platforms; and 3) applications that need on-the-fly multisensory data such as on-policy reinforcement learning.

In summary, our main contributions are as follows: First, we introduce a new large multisensory dataset of 3D objects in the form of implicit neural representations, which is 10 times larger in scale compared to existing work. We significantly improve the multisensory rendering quality for vision, audio, and touch, while being orders of magnitude faster in rendering time. Second, we show that learning with our virtualized objects can successfully transfer to a series of real-world tasks, offering a new path and testbed for multisensory learning for computer vision and robotics.

2. Related Work

Object datasets. Objects are modeled in different ways across different datasets. Image datasets such as ImageNet [15] and MS COCO [39] model objects in 2D. Datasets of synthetic 3D CAD models such as ModelNet [72] and ShapeNet [11] focus on the geometry of objects without modeling their realistic visual textures. Pix3D [66], IKEA Objects [38], and Object3D [73] align 3D CAD models to objects in real images, but they are either limited in size or make unignorable approximations in the 2D-3D alignment. BigBIRD [62] and YCB [10] directly model real-world objects but only for a small number of object instances. ABO [14] was recently introduced, containing 3D models for over 8K objects of real household objects, but it focuses only on the visual modality, similar to the other datasets above.

[1]Throughout, we refer the OBJECTFOLDER 1.0 [18] dataset as 1.0 and our dataset as 2.0 for convenience.

Alternatively, OBJECTFOLDER 2.0 contains 1,000 3D objects in the form of implicit neural representations, each of which encodes realistic visual, acoustic, and tactile sensory data for the corresponding object. Compared to OBJECTFOLDER 1.0 [18], our dataset is not only 10 times larger in the amount of objects, but also we significantly improve the quality of the multisensory data while being 100 times faster in rendering time. Furthermore, while OBJECTFOLDER 1.0 only performs tasks in simulation, we show that learning with our virtualized objects generalizes to the objects' real-world counterparts.

Implicit neural representations. Coordinate-based multilayer perceptrons (MLPs) have attracted much attention lately and have been used as a new way to parameterize different types of natural signals. They are used to learn priors over shapes [12, 44, 54]; represent the appearance of static scenes [45, 64], dynamic scenes [49, 55], or individual objects [25, 48]; and even encode other non-visual modalities such as wavefields, sounds, and tactile signals [18, 63].

We also use MLPs to encode object-centric visual, acoustic, and tactile data similar to [18], but our new object-centric implicit neural representations encode the intrinsics of objects more realistically and flexibly. Furthermore, inspired by recent techniques [23, 28, 40, 42, 47, 57, 74] on speeding up neural volume rendering [32], we largely reduce the rendering time of visual appearance, making inference of all sensory modalities real-time.

Multisensory learning. A growing body of work leverages other sensory modalities as learning signals in addition to vision, with audio and touch being the most popular. For audio-visual learning, inspiring recent work integrates sound and vision for a series of interesting tasks, including self-supervised representation learning [33, 50, 51], audio-visual source separation [17, 19, 21, 77], sound localization in video frames [60, 68], visually-guided audio generation [20, 46], and action recognition [22, 71]. For visuo-tactile learning, the two sensory modalities are used for cross-modal prediction [37] and representation learning [36, 56]. Touch is also used to augment vision for 3D shape reconstruction [65, 67], robotic grasping [8, 9], and object contact localization [43]. Earlier work on modeling multisensory physical behavior of 3D objects [53] proposes a system to directly measure contact textures and sounds, but mainly for the purpose of better modeling virtual object interaction and creating animations.

OBJECTFOLDER 2.0 is a potential testbed for various multisensory learning tasks involving all three modalities. Different from the works above, instead of learning with certain sensory modalities for a particular task, our goal is to introduce a dataset of implicitly represented objects with realistic visual, acoustic, and tactile sensory data, making multisensory learning easily accessible to the computer vision and robotics community.

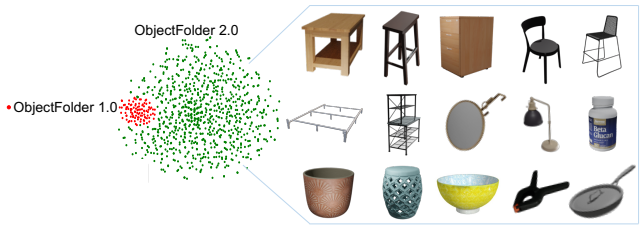


Figure 2. Example objects in OBJECTFOLDER 2.0. Each dot on the left represents an object in our dataset with red dots representing objects from OBJECTFOLDER 1.0.

3. A Large Repository of Diverse Objects

OBJECTFOLDER 2.0 contains 1,000 3D objects in the form of implicit neural representations. Among the 1,000 objects, we use all 100 objects from OBJECTFOLDER 1.0 [18], which consists of high quality 3D objects from 3D Model Haven [1], YCB [10], and Google Scanned Objects [2]. The recently introduced ABO dataset [14] is another rich repository of real-world 3D objects, containing about 8K object models with high-quality 3D meshes, which come from Amazon.com product listings. For each object, we obtain metadata such as category, material, color, and dimensions on the real product's publicly available webpage. We filter the dataset by material type and only keep objects of the following materials: ceramic, glass, wood, plastic, iron, polycarbonate, and steel. We visually inspect each object's product images to make sure the metadata is correct and keep the object if its material property is approximately homogeneous. These steps ensure that the selected objects are acoustically simulatable as will be described in Sec. 4.2. In the end, we obtain 855 objects from the ABO dataset. Additionally, we obtain 45 objects of polycarbonate material type from Google Scanned Objects.

Fig. 2 shows some example objects in our dataset.[2] OBJECTFOLDER 2.0 is an order of magnitude larger than OBJECTFOLDER 1.0 and contains common household items of diverse categories including wood desks, ceramic bowls, plastic toys, steel forks, glass mirrors, etc.

4. Improved Multisensory Simulation and Implicit Representations

We propose a new simulation pipeline to obtain the multisensory data based on the objects' physical properties. Each object is represented by an *Object File*, which is an implicit neural representation network that encodes the complete multisensory profile of the object. See Fig. 1. Implicit representations have many advantages compared to conventional signal representations, which are usually discrete. We can parameterize each sensory modality as a continuous function that maps from some extrinsic parameters (e.g., camera view point and lighting conditions for vision, impact strength for audio, gel deformation for touch) to the corresponding sensory signal at a certain location or condition. Implicit neural representations serve as an approximation to this continuous function via a neural network. This makes the memory required to store the original sensory data independent of those extrinsic parameters, allowing the implicit representations to be easily streamed to users. Furthermore, thanks to the continuous property of implicit neural representations, the sensory data can be sampled at arbitrary resolutions.

Each *Object File* has three sub-networks: VisionNet, AudioNet, and TouchNet (see Fig. 3). In the following, we introduce the details of how we simulate the three modalities and how we use multi-layer perceptrons (MLPs) to encode the data.

4.1. Vision

Background. Recent work [25] proposes to represent the appearance of each object by a neural network F_v that models the object-centric neural scattering function (OSF). F_v takes as input a 3D location $\mathbf{x} = (x, y, z)$ in the object coordinate frame and the lighting condition at that location (ω_i, ω_o), where $\omega_i = (\phi_i, \theta_i)$ and $\omega_o = (\phi_o, \theta_o)$ denote the incoming and outgoing light directions, respectively. The output is the volume density σ and fraction of the incoming light that is scattered in the outgoing direction $\rho = (\rho_r, \rho_g, \rho_b)$. The amount of light scattered at a point $\mathbf{x}$ can be obtained as follows:

$$L_s(\mathbf{x}, \omega_\mathbf{o}) = \int_S L(\mathbf{x}, \omega_i) f_\rho(\mathbf{x}, \omega_\mathbf{i}, \omega_\mathbf{o}) d\omega_\mathbf{i}, \qquad (1)$$

where S is a unit sphere, $L(\mathbf{x}, \omega_i)$ denotes the amount of light scattered at point $\mathbf{x}$ along direction ω_i, and f_ρ evaluates the fraction of light incoming from direction $\omega_\mathbf{i}$ at the point that scatters out in direction $\omega_\mathbf{o}$.

Classic volume rendering [32] is then used to render the color of any ray passing through the object. To render a single image pixel, a ray is cast from the camera's eye through the pixel's center. We denote the direction of the camera ray as $\mathbf{r}(t) = \mathbf{x}_0 + t\omega_\mathbf{o}$. A number of points $\mathbf{x}_1, \mathbf{x}_\mathbf{2}, \ldots, \mathbf{x}_K$ are sampled along the ray. The final expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t)$ can be obtained by α-blending the list of K color values $(L_s(\mathbf{x}_1, \omega_\mathbf{o}), L_s(\mathbf{x}_2, \omega_\mathbf{o}), \ldots, L_s(\mathbf{x}_K, \omega_\mathbf{o}))$ with the following equation:

$$C(\mathbf{r}) = \sum_{i=1}^{K} T_i (1 - \exp(-\sigma_i \delta_i)) L_s(\mathbf{x}_i, \omega_\mathbf{o}), \qquad (2)$$

where $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$ denotes the accumulated transmittance along the ray, and $\delta_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|$ denotes the distance between two adjacent sample points.

[2]Note that the original object meshes we use in our dataset all come from prior datasets [2, 10, 14], and our contribution is a pipeline to create multisensory object assets based on these mesh models.

KiloOSF. The above process has to be repeated for every pixel to render an image. Due to the large number of required forward passes through F_v, this makes it very time-consuming even on high-end consumer GPUs.

Inspired by many recent works on speeding up neural rendering [23, 28, 47, 57, 74], we build upon KiloNeRF [57] and introduce KiloOSF as our VisionNet. Instead of using a single MLP to represent the entire scene, KiloNeRF represents the static scene with a large number of independent and small MLPs. Each individual MLP is assigned a small portion of the scene, making each small network sufficient for photo-realistic rendering.

Similarly, we subdivide each *object* into a uniform grid of resolution $\mathbf{s} = (s_x, s_y, s_z)$ with each grid cell of 3D index $\mathbf{i} = (i_x, i_y, i_z)$. Then we define a mapping m from position $\mathbf{x}$ to index $\mathbf{i}$ through the following spatial binning:

$$m(\mathbf{x}) = \lfloor (\mathbf{x} - \mathbf{b}_{\text{min}}) / ((\mathbf{b}_{\text{max}} - \mathbf{b}_{\text{min}})/\mathbf{s}) \rfloor, \tag{3}$$

where $\mathbf{b}_{\text{min}}$ and $\mathbf{b}_{\text{max}}$ are the respective minimum and maximum bounds of the axis aligned bounding box (AABB) enclosing the object. For each grid cell, a tiny MLP network with parameters $v(\mathbf{i})$ is used to represent the corresponding portion of the object. Then, the color and density values at a point $\mathbf{x}$ and direction $\mathbf{r}$ can be obtained by first determining the index $m(\mathbf{x})$ responsible for the grid cell that contains this point, then querying the respective tiny MLP:

$$(\mathbf{c}, \sigma) = F_{v(m(\mathbf{x}))}(\mathbf{x}, \mathbf{r}). \tag{4}$$

Following KiloNeRF [57], we use a "training with distillation" strategy to avoid artifacts in rendering. We first train an ordinary OSF [25] model for each object and then distill the knowledge of the teacher model into the KiloOSF model. We also use empty space skipping and early ray termination to increase rendering efficiency. See [57] for details. Compared with OBJECTFOLDER 1.0, our new VisionNet design significantly accelerates the rendering process at inference time 60 times (see Table 1) while simultaneously achieving better visual rendering quality.

4.2. Audio

Background. Linear modal analysis is a standard way to perform physics-based 3D modal sound synthesis [31, 58, 69]. A 3D linear elastic dynamic system can be modeled with the following linear deformation equation:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}, \tag{5}$$

where $\mathbf{x}$ denotes the nodal displacement, and $\mathbf{M}$, $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$, $\mathbf{K}$ represent the mass, Rayleigh damping, and stiffness matrices, respectively.[3] $\mathbf{f}$ represents the external nodal force applied to the object that stimulates the vibration. Through generalized eigenvalue decomposition $\mathbf{KU} = \mathbf{\Lambda M U}$, the above equation can be reformulated into the following form:

$$\ddot{\mathbf{q}} + (\alpha\mathbf{I} + \beta\mathbf{\Lambda})\dot{\mathbf{q}} + \mathbf{\Lambda}\mathbf{q} = \mathbf{U}^{\mathbf{T}}\mathbf{f}, \tag{6}$$

where $\mathbf{\Lambda}$ is a diagonal matrix, and $\mathbf{q}$ satisfies $\mathbf{x} = \mathbf{U}\mathbf{q}$. The solution to the above equation is N damped sinusoidal waves, each representing a mode signal. The i_{th} mode is

$$q_i = g_i e^{-d_i t} \sin(2\pi\omega_i t),\ i = \{1, 2, \ldots, N\} \tag{7}$$

where ω_i, d_i, and g_i represent the damped natural frequencies, damping coefficients, and gains of the modes signals, respectively. Note that the gains g_i of each mode are specific to the contact force and the contact location on the object, while the frequencies ω_i and damping coefficients d_i of each mode are intrinsic parameters of the object.

AudioNet. We convert the surface mesh of each object into a volumetric quadratic tedrahedral mesh using a sequential approach designed for object meshes from the wild [29], then use Finite Element Methods (FEM) [30] on the resultant tetrahedral mesh with second-order elements in Abaqus [5] to perform the modal analysis process described above. We simulate the vibration modes from contacting each vertex on the tedrahedral mesh with unit force in each axis direction. Then, we train an MLP that takes the vertex coordinate of the tedrahedral mesh as input and predicts the vector of gains of each mode for that vertex when contacted by unit force for each axis direction.

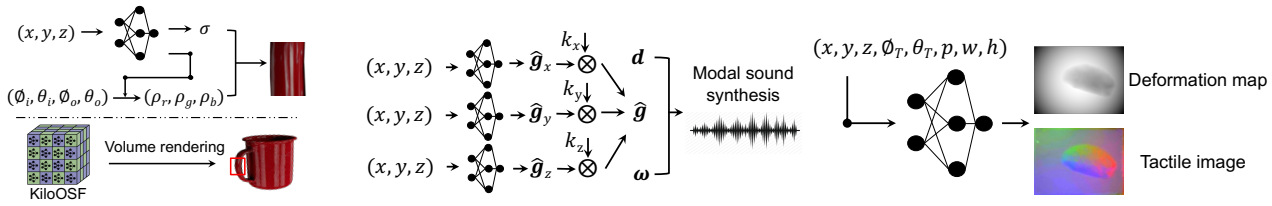
At inference time, an object's impulse response can be predicted by first using the network to predict the gains g_i of each mode, then constructing the response by summing the exponentially decaying sinusoids parameterized by the gains $\hat{g}_i$ predicted from the network, along with the frequencies ω_i and dampings d_i obtained from modal analysis. We decompose the external force $\mathbf{f}$ at a vertex into a linear combination of unit forces along the three orthogonal axis directions: $\mathbf{f} = k_x\mathbf{f}_x + k_y\mathbf{f}_y + k_z\mathbf{f}_z$. The predicted gains $\hat{\mathbf{g}}$ excited by $\mathbf{f}$ can be obtained as follows: $\hat{\mathbf{g}} = k_x\hat{\mathbf{g}}_x + k_y\hat{\mathbf{g}}_y + k_z\hat{\mathbf{g}}_z$, where $\hat{\mathbf{g}}_x, \hat{\mathbf{g}}_y, \hat{\mathbf{g}}_z$ denote the the respective gains obtained from the three branches of AudioNet. Finally, combining the frequencies $\boldsymbol{\omega}$ and damping coefficients $\mathbf{d}$, we synthesize the audio waveform:

$$S(t) = \sum_{i=1}^{N} \hat{g}_i e^{-d_i t} \sin(2\pi\omega_i t), \tag{8}$$

where $\hat{g}_i$, d_i, and ω_i represent elements of $\hat{\mathbf{g}}$, $\mathbf{d}$, and $\boldsymbol{\omega}$, respectively.

As opposed to using a volumetric hexahedron mesh for modal analysis as in OBJECTFOLDER 1.0, the higher-order tetrahedral meshes we use for modal analysis capture finer features and surface curvature as well as more precise elastic deformations, at the same representation size. Thus it can more accurately model the acoustic properties of the objects [7, 27, 59]. Moreover, the AudioNet in 1.0 directly predicts a complex audio spectrogram, which is of much higher

[3] The values of these matrices depend on the object's scale and material. See Supp. for the mapping from material type to material parameters.



(a) VisionNet (b) AudioNet (c) TouchNet

Figure 3. Each *Object File* implicit neural representation network contains three sub-networks: VisionNet, AudioNet, and TouchNet. Compared with OBJECTFOLDER 1.0, we greatly accelerate VisionNet inference by representing each object with thousands of individual MLPs; for AudioNet, we only predict the parts of the signal that are location-dependent instead of directly predicting the audio spectrograms, which significantly improves the rendering quality and also accelerates inference; our new TouchNet can render tactile readings of varied rotation angles and gel deformations, whereas only a single tactile image can be rendered per vertex in 1.0.

	Vision	Audio	Touch	Total
OBJECTFOLDER 1.0 [18]	3.699	0.420	0.010	4.129
OBJECTFOLDER 2.0 (Ours)	0.062	0.035	0.014	0.111

Table 1. Time comparison for rendering one observation sample for each modality, in seconds.

dimension and is limited to a fixed resolution and temporal length. We instead only predict the parts of the signal that are location-dependent, and then analytically obtain the remainder of the modes signal. This significantly improves the quality of audio rendering with our new implicit representation network. See Table 2 and Fig. 4 for a comparison.

4.3. Touch

Background. We use the geometric measurement from a GelSight tactile sensor [16, 75] as the tactile reading. GelSight is a vision-based tactile sensor that interacts the object with an elastomer and measures the geometry of the contact surface with an embedded camera. It has a very high spatial resolution of up to 25 micrometers and can potentially be used to synthesize readings from other tactile sensors [35, 52]. To simulate tactile sensing with GelSight, we need to simulate both the deformation of the contact and the optical response to the deformation. For our tactile simulation, we aim to achieve the following three goals: 1) Being flexible to render tactile readings for touches of varied location, orientation, and pressing depth; 2) Being fast to efficiently render data for training TouchNet; 3) Being realistic to generalize to real-world touch sensors.

TouchNet. To achieve the three goals above, we adopt a two-stage approach to render realistic tactile signals. First, we simulate the contact deformation map, which is constructed from the object's shape in the contact area and the gelpad's shape in the non-contact area to represent the local shape at the point of contact. We simulate the sensor-object interaction with Pyrender [4] to render deformation maps using OpenGL [3] with GPU-acceleration, reaching 700 fps for data generation.

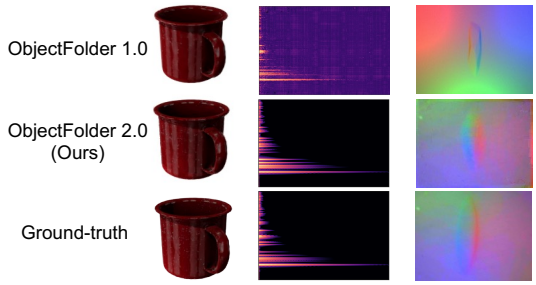


Figure 4. Comparing the visual, acoustic, and tactile data rendered from OBJECTFOLDER 1.0, OBJECTFOLDER 2.0 (Ours), and the corresponding ground-truth simulations for the YCB mug. See Supp. for more examples.

We design TouchNet to encode the deformation maps from contacting each vertex on the object. We represent the tactile readings of each object as an 8D function whose input is a 3D location $\mathbf{x} = (x, y, z)$ in the object coordinate frame, a 3D unit contact orientation parametrized as (θ_T, ϕ_T), gel penetration depth p, and the spatial location (w, h) in the deformation map. The output is the per-pixel value of the deformation map for the contact. TouchNet models this continuous function as an MLP network $F_T : (x, y, z, \theta_T, \phi_T, p, w, h) \longrightarrow d$ that maps each input 8D coordinate to its corresponding value in the deformation map. After rendering the deformation map, we utilize the state-of-the-art GelSight simulation framework—Taxim [61], an example-based tactile simulation model that is calibrated with a real GelSight sensor, to render tactile RGB images from the deformation maps.

Compared to the TouchNet in OBJECTFOLDER 1.0, which can only render a single tactile image along the vertex normal direction per vertex, our new design of TouchNet can generate tactile outputs for rotation angles within $\pm 15^\circ$ and pressing depth in the range of 0.5-2 mm. Furthermore, with the help of Taxim, the mapping from the deformation maps to the tactile optical outputs can be easily calibrated to different real vision-based tactile sensors, producing realistic tactile optical outputs that enable Sim2Real transfer.

	Vision		Audio		Touch	
	PSNR ↑	SSIM ↑	STFT Distance ($\times 10^{-5}$) ↓	ENV Distance ($\times 10^{-4}$) ↓	PSNR ↑	SSIM ↑
OBJECTFOLDER 1.0 [18]	35.7	0.97	4.94	7.65	27.9	0.64
OBJECTFOLDER 2.0 (Ours)	36.3	0.98	0.19	1.29	31.6	0.78

Table 2. Comparing with OBJECTFOLDER 1.0 on the multisensory data rendering quality. ↓ lower better, ↑ higher better.

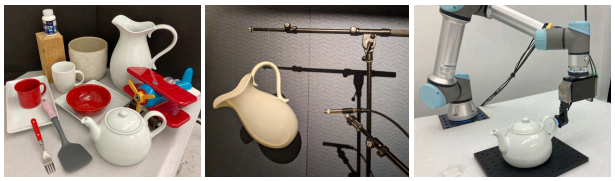


Figure 5. Illustration of real-world objects used in experiments and our hardware set-up for collecting real-world impact sounds and tactile data.

4.4. OBJECTFOLDER 1.0 vs. OBJECTFOLDER 2.0

OBJECTFOLDER 2.0 significantly advances OBJECTFOLDER 1.0 in multisensory simulation and the design of implicit neural representations. Table 1 shows the rendering time comparison. Our new network design is orders of magnitude faster compared to OBJECTFOLDER 1.0, making rendering of all three sensory modalities real-time. The rendering quality is also greatly improved, especially for audio and touch as shown in the example of Fig. 4. Our KiloOSF VisionNet renders images that match the ground-truth well while being $60\times$ faster than OBJECTFOLDER 1.0. While directly predicting audio spectrograms cannot capture the details of the modes signal and leads to artifacts in the background, our AudioNet renders audio in a much more accurate manner. For touch, to make a fair comparison, we use the TACTO [70] simulation used in OBJECTFOLDER 1.0 and the tactile readings from real-world GelSight sensors as the ground truth instead. Our TouchNet output matches well with the real tactile readings.

Table 2 shows the quantitative comparisons. For visual and tactile rendering, we compare using standard metrics: peak signal-to-noise ratio (PSNR) and structural index similarity (SSIM) between the rendered image and the ground-truth image. For audio rendering, we report the STFT distance, which is the euclidean distance between the spectrograms of the ground-truth and the predicted modes signals, and the Envelope (ENV) Distance, which measures the Euclidean distance between the envelopes of the ground-truth and the predicted modes signals. For touch, because OBJECTFOLDER 1.0 uses the DIGIT [35] tactile sensor, we compare with the real tactile images collected from a DIGIT sensor and a GelSight sensor for 1.0 and ours, respectively. Our TouchNet based on GelSight sensors has a smaller Sim2Real gap.

5. Sim2Real Object Transfer

The goal of building OBJECTFOLDER 2.0 is to enable generalization to real-world objects by learning with the virtual objects from our dataset. We demonstrate the utility of the dataset by evaluating on three tasks including object scale estimation, contact localization, and shape reconstruction. In each task, we transfer the models learned on OBJECTFOLDER 2.0 to real-world objects. See Fig. 5 for an illustration of the 13 objects used in our experiments, and the hardware set-up for collecting real impact sounds and GelSight tactile readings.

5.1. Object Scale Estimation

All sensory modalities of objects are closely related to their scales. We want to demonstrate that learning with our virtualized objects can successfully transfer to scale estimation for a real object based on either its visual appearance, an impact sound, or a sequence of tactile readings. We train on the rendered multisensory data from our dataset, and test on 8 real objects from which we have collected real-world sensory data for all three modalities.

For vision and audio, we train ResNet-18 [26] that takes either an RGB image of the object or the magnitude spectrogram of an impact sound as input to predict object scale[4]. From a single local tactile reading, it is almost impossible to predict the scale of the object. Therefore, we use a recurrent neural network to combine features from 10 consecutive touch readings for tactile-based scale prediction. See Supp. for details.

Table 3 shows the results. "Random" denotes the baseline that randomly predicts a scale value within the same range as our models. We compare with models trained on sensory data from OBJECTFOLDER 1.0. Both OBJECTFOLDER 1.0 and our dataset achieve high scale prediction accuracy on virtual objects. However, models trained on our multisensory data generalize much better to real-world objects, demonstrating the realism of our simulation and accurate encoding of our implicit representation networks. Among the three modalities, tactile data has the smallest Sim2Real gap compared to vision and audio.

5.2. Tactile-Audio Contact Localization

When interacting with an object of known shape, accurately identifying the location where the interaction happens

[4]We define the scale of an object as the length of the longest side of the axis aligned bounding box (AABB) enclosing the object.

		Virtual Objects	Real Objects
	Random	14.5	14.5
1.0 [18]	Vision	0.80	7.41
	Audio	0.57	6.85
	Touch	0.19	4.92
2.0 (Ours)	Vision	0.79	5.08
	Audio	0.20	4.68
	Touch	0.45	3.51

Table 3. Results on object scale prediction. We report the average difference between the predicted and the ground-truth scales of the objects in centimeters.

is of great practical interest. Touch gives local information about the contact location, and impact at varied surface locations produces different modal gains for the excited sound. We investigate the potential of using the impact sounds and/or the tactile readings associated with the interaction for contact localization.

We apply particle filtering [41] to localize the sequence of contact locations from which tactile readings or impact sounds are collected. Particle filters are used to estimate the posterior density of a latent variable given observations. Here, observations are either tactile sensor readings when touching the object or impact sounds excited at the contact locations. The latent variable is the current contact location on the object's surface. For touch, we extract features from an FCRN network [34] pre-trained for depth prediction from tactile images. For audio, we extract MFCC features from each 3s impact sound. We compare these features with particles sampled from the object surfaces that represent the candidate contact locations. Particles with high similarity scores to the features of the actual tactile sensor reading or impact sound are considered more likely to be the true contact location. In each iteration, we weight and re-sample the particles based on the similarity scores, and then update the particles' locations based on the relative translations between two consecutive contacts obtained from the robot end-effector. We choose the 10 particles with the highest similarity scores as the candidate contact locations. For each object, we iterate the above process for 5-7 times until the predicted current contact location converges to a single location on the object's surface. We perform experiments both in simulation and in real world.

Table 4 shows the results for six objects of complex shapes. We use the mean Euclidean distance with respect to the ground-truth contact location as the evaluation metric similar to [6]. We compare the localization accuracy for using only touch readings, impact sounds, or their combinations, and a baseline that randomly predicts a surface position as the contact location. We can see that touch-based contact location is much more accurate than using audio.

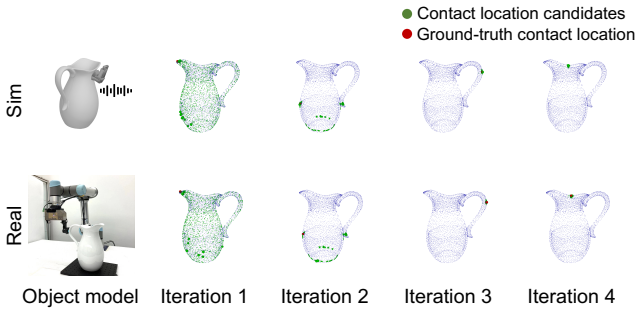


Figure 6. Qualitative results for contact localization with touch readings and impact sounds. Top: in simulation, bottom: real-world experiments. The candidate contact locations are shown as green particles in the particle filter. After several iterations shown from left to right in each row, the green particles converge to the ground-truth contact location shown as the red particle.

Combining the two modalities leads to the best Sim2Real performance. Fig. 6 shows a qualitative example for tactile-audio contact location with the pitcher object.

5.3. Visuo-Tactile Shape Reconstruction

Single-image shape reconstruction has been widely studied in the vision community [11, 13, 44, 54]. However, in cases where there is occlusion such as during dexterous manipulation, tactile signals become valuable for perceiving the shape of the objects. Vision provides coarse global context, while touch offers precise local geometry. Here, we train models to reconstruct the shape of 3D objects from a single RGB image containing the object and/or a sequence of tactile readings on the object's surface.

We use Point Completion Network (PCN) [76], a learning-based approach for shape completion, as a testbed for this task. For touch, we use 32 tactile readings and map the associated deformation maps to a sparse point cloud given the corresponding touching poses. The sparse point cloud is used as input to the PCN network for generating a dense and complete point cloud. For vision, instead of using a series of local contact maps as partial observations of the object, a global feature extracted from a ResNet-18 network from a single image containing the object is used to supervise the shape completion process. For shape reconstruction with vision and touch, we use a two-stream network that merges the predicted point clouds from both modalities with a fully-connected layer to predict the final dense point cloud. See Supp. for details.

Table 5 shows the results for six objects of different shapes. Compared to the "Average" baseline that uses the average ground-truth mesh of the 6 objects as the prediction, shape reconstructions from a single image and a sequence of touch readings perform much better. Combining the geometric cues from both modalities usually leads to the best Sim2Real transfer performance. Fig. 7 shows some

Modalities												
	Sim	Real	Sim	Real	Sim	Real	Sim	Real	Sim	Real	Sim	Real
Random	6.74	6.74	12.96	12.96	4.28	4.28	9.39	9.39	14.53	14.53	14.21	14.21
Audio	1.88	1.79	0.26	1.16	0.65	4.67	0.23	1.04	0.14	-	0.74	-
Touch	0.04	1.26	0.03	0.78	0.18	1.30	0.04	0.44	0.04	0.91	0.04	3.82
Audio + Touch	0.02	0.59	0.04	0.36	0.09	0.51	0.04	0.63	0.23	-	0.30	-

Table 4. Results on audio-tactile contact localization. We report the mean distance w.r.t. the ground-truth contact locations in centimeters.

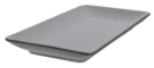
Modalities												
	Sim	Real	Sim	Real	Sim	Real	Sim	Real	Sim	Real	Sim	Real
Average	2.12	2.01	2.97	1.91	4.80	3.26	4.53	4.49	2.44	2.53	2.52	3.29
Vision	0.25	0.32	0.30	0.72	0.51	0.74	0.38	0.66	0.32	0.40	0.49	0.99
Touch	0.24	0.56	0.29	0.80	0.35	0.61	0.38	0.43	0.30	0.41	0.36	1.11
Vision + Touch	0.09	0.25	0.18	0.46	0.26	0.43	0.24	0.32	0.18	0.24	0.23	1.20

Table 5. Results on visuo-tactile shape reconstruction. We report the Chamfer-L1 distance w.r.t. the ground-truth meshes in centimeters.

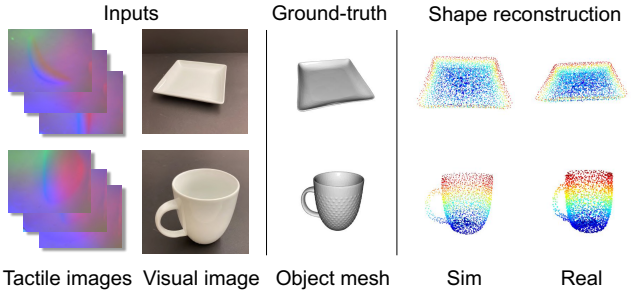


Figure 7. Qualitative results for visual-tactile shape reconstruction in simulation (Sim) and real-world (Real) for the square tray and the coffee mug.

qualitative results for shape reconstruction with vision and touch. We can see that the predicted point clouds in both simulation and real-world experiments accurately capture the shapes of the two objects, and matches the ground-truth object meshes well.

6. Broader Impact and Limitations

We will release our dataset and code upon publication of the paper, so that it can be easily accessible to the community as a standard benchmark for multisensory learning. This avoids the need to purchase real-world objects for such tasks, and can especially benefit people in areas where international shipping and purchasing of specific real-world objects is challenging. Furthermore, our implicit representation is computationally much cheaper to render multisensory data compared to the initial multisensory simulation, which is potentially more environmentally friendly.

Bridging the gap between sim and real for multisensory object-centric learning is inherently difficult. While we have shown Sim2Real transfer for a series of objects, the objects in our dataset are all rigid-body objects, and we assume single homogeneous material for the whole object. However, real-world objects are complex and often contain several parts, which can be non-rigid and are of different material types. Furthermore, the 3D space in which these objects are located is of diverse lighting/noise conditions, reverberation effects, etc. Sim2Real object transfer is challenging without modeling all these factors, which we leave as future work.

7. Conclusion

OBJECTFOLDER 2.0 is a dataset of 1,000 objects in the form of implicit neural representations aimed at advancing multisensory learning in computer vision and robotics. Compared to existing work, our dataset is 10 times larger in scale and orders of magnitude faster in rendering time. We also significantly improve the quality and realism of the multisensory data. We show that models learned with our virtualized objects successfully transfer to their real-world counterparts on three challenging tasks. Our dataset offers a promising path for multisensory object-centric learning in computer vision and robotics, and we look forward to the research that will be enabled by OBJECTFOLDER 2.0.

Acknowledgements. We thank Sudharshan Suresh, Mark Rau, Doug James, and Stephen Tian for helpful discussions. The work is in part supported by the Stanford Institute for Human-Centered AI (HAI), the Stanford Center for Integrated Facility Engineering, NSF CCRI #2120095, Toyota Research Institute (TRI), Samsung, Autodesk, Amazon, Adobe, Google, and Facebook.

References

[1] 3D Model Haven. https://3dmodelhaven.com/. 2, 3

[2] Google Scanned Objects. https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/Google%20Scanned%20Objects. 2, 3

[3] OpenGL. https://www.opengl.org. 5

[4] Pyrender. https://github.com/mmatl/pyrender. 5

[5] FEA Abaqus et al. Dassault systemes simulia corporation. 2021. 4

[6] Maria Bauza, Eric Valls, Bryan Lim, Theo Sechopoulos, and Alberto Rodriguez. Tactile object pose estimation from the first touch with geometric contact rendering. *arXiv preprint arXiv:2012.05205*, 2020. 7

[7] Gaurav Bharaj, David IW Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. Computational design of metallophone contact sounds. *ToG*, 2015. 4

[8] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *RA-L*, 2018. 2

[9] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In *CoRL*, 2017. 2

[10] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *ICRA*, 2015. 2, 3

[11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2, 7

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 2

[13] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 7

[14] Jasmine Collins, Shubham Goel, Achleshwar Luthra, Leon Xu, Kenan Deng, Xi Zhang, Tomas F Yago Vicente, Himanshu Arora, Thomas Dideriksen, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. *arXiv preprint arXiv:2110.06199*, 2021. 2, 3

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2

[16] Siyuan Dong, Wenzhen Yuan, and Edward H Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *IROS*, 2017. 5

[17] Chuang Gan, Deng Huang, Hang Zhao, Joshua B Tenenbaum, and Antonio Torralba. Music gesture for visual sound separation. In *CVPR*, 2020. 2

[18] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021. 2, 3, 5, 6, 7

[19] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018. 2

[20] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *CVPR*, 2019. 2

[21] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *ICCV*, 2019. 2

[22] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020. 2

[23] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *arXiv preprint arXiv:2103.10380*, 2021. 2, 4

[24] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[25] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2, 3, 4

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[27] ZC He, GY Li, ZH Zhong, AG Cheng, GY Zhang, Eric Li, and GR Liu. An es-fem for accurate analysis of 3d mid-frequency acoustics using tetrahedron mesh. *Computers & Structures*, 2012. 4

[28] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021. 2, 4

[29] Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. Tetrahedral meshing in the wild. *ToG*, 2018. 4

[30] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. 2012. 4

[31] Xutong Jin, Sheng Li, Tianshu Qu, Dinesh Manocha, and Guoping Wang. Deep-modal: real-time impact sound synthesis for arbitrary shapes. In *ACMMM*, 2020. 4

[32] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *SIGGRAPH*, 1984. 2, 3

[33] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-training of audio and video representations from self-supervised temporal synchronization. In *NeurIPS*, 2018. 2

[34] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 7

[35] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al.

Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *RA-L*, 2020. 5, 6

[36] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *ICRA*, 2019. 2

[37] Yunzhu Li, Jun-Yan Zhu, Russ Tedrake, and Antonio Torralba. Connecting touch and vision via cross-modal prediction. In *CVPR*, 2019. 2

[38] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013. 2

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2

[40] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 2

[41] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 1998. 7

[42] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 2

[43] Shan Luo, Wenxuan Mou, Kaspar Althoefer, and Hongbin Liu. Localizing the object contact through matching tactile features with visual map. In *ICRA*, 2015. 2

[44] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2, 7

[45] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[46] Pedro Morgado, Nono Vasconcelos, Timothy Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360° video. In *NeurIPS*, 2018. 2

[47] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. 2021. 2, 4

[48] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[49] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019. 2

[50] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. 2

[51] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 2

[52] Akhil Padmanabha, Frederik Ebert, Stephen Tian, Roberto Calandra, Chelsea Finn, and Sergey Levine. Omnitact: A multi-directional high-resolution touch sensor. In *ICRA*, 2020. 5

[53] Dinesh K Pai, Kees van den Doel, Doug L James, Jochen Lang, John E Lloyd, Joshua L Richmond, and Som H Yau. Scanning physical interaction behavior of 3d objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001. 2

[54] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2, 7

[55] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021. 2

[56] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *ECCV*, 2016. 2

[57] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. 2021. 2, 4

[58] Zhimin Ren, Hengchin Yeh, and Ming C Lin. Example-guided physically based modal sound synthesis. *TOG*, 2013. 4

[59] Teseo Schneider, Yixin Hu, Xifeng Gao, Jeremie Dumas, Denis Zorin, and Daniele Panozzo. A large scale comparison of tetrahedral and hexahedral elements for finite element analysis. *arXiv preprint arXiv:1903.09332*, 2019. 4

[60] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *CVPR*, 2018. 2

[61] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *arXiv preprint arXiv:2109.04027*, 2021. 5

[62] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird:(big) berkeley instance recognition dataset. In *ICRA*, 2014. 2

[63] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 2

[64] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[65] Edward J Smith, Roberto Calandra, Adriana Romero, Georgia Gkioxari, David Meger, Jitendra Malik, and Michal Drozdzal. 3d shape reconstruction from vision and touch. In *NeurIPS*, 2020. 2

[66] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 2

[67] Sudharshan Suresh, Zilin Si, Joshua G Mangelson, Wenzhen Yuan, and Michael Kaess. Efficient shape mapping through

dense touch and vision. *arXiv preprint arXiv:2109.09884*, 2021. 2

[68] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 2

[69] Jui-Hsien Wang and Doug L James. Kleinpat: optimal mode conflation for time-domain precomputation of acoustic transfer. In *SIGGRAPH*, 2019. 4

[70] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible and open-source simulator for high-resolution vision-based tactile sensors. *arXiv preprint arXiv:2012.08456*, 2020. 6

[71] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. Multi-stream multi-class fusion of deep networks for video classification. In *ACMMM*, 2016. 2

[72] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 1, 2

[73] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *ECCV*, 2016. 2

[74] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2, 4

[75] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 2017. 5

[76] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, 2018. 7

[77] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018. 2